6. Juana

Program Name: Juana.java **Input File: juana.dat**

Juana has been working with 2-dimension tables of data and would like to extract an arbitrary “chunk” of the data. She will provide the number of rows and columns in the original table along with the data to populate the table. She will then provide a starting point by identifying the row and column positions of the top left corner of the “chunk” along with the numbers of columns and rows desired. Juana does not have programming experience so her tables start with row 1 and column 1 at the top-left corner. She has provided the following example:

	Col 1	Col 2	Col 3	Col 4	Col 5
Row 1	1	2	3	4	5
Row 2	6	7	8	9	10
Row 3	11	12	13	14	15
Row 4	16	17	18	19	20

The table contains 4 rows and 5 columns with the data as shown above. Juana would like to extract the “chunk” of data that starts at row 3 and column 2. She wants 3 columns across the rows and 2 total rows as shown by the shading above.

Can you create a “chunk” extraction program for Juana?

Input: First line of data file contains a positive integer T, the number of test cases that follow with 1 ≤ T ≤ 10. Each test case starts with a line containing the number of rows R and columns C for the table with 2 ≤ R,C ≤ 15. That line will then be followed by R lines of data each containing C integers in the range [0, 1000). The data items will be right-aligned with leading spaces for student viewing below. The next line will contain 4 integers separated by a space. The first pair of integers are the row and column numbers of the top-left corner. The next 2 integers are the count of columns and rows desired in the “chunk”. All 4 integers will have the same range as specified for R and C above. There is no guarantee that Juana specifies the starting point and sizes for the “chunk” correctly. When there is not enough data to extract the complete “chunk” an error message will be displayed instead of the requested “chunk”.

Output: For each test case, display a line with the test case number formatted as shown in the sample. If the “chunk” cannot be extracted, display the error message as shown below on the same line as the test case. Otherwise, the extracted “chunk” of data is displayed below the test case line. Each data item is right-aligned in a column that is 5 positions wide. Follow the “chunk” with a line containing 20 hyphens “--------------------”.

~ Sample input and output on next page ~

Juana, continued

Sample input:

```
3
4 5
    1     2     3     4     5
    6     7     8     9    10
   11    12    13    14    15
   16    17    18    19    20
3 2 3 2
7 6
    1     2     3     4     5     6
    7     8     9    10    11    12
   13    14    15    16    17    18
   19    20    21    22    23    24
   25    26    27    28    29    30
   31    32    33    34    35    36
   37    38    39    40    41    42
5 4 2 3
3 3
    1     2     3
    4     5     6
    7     8     8
2 2 3 1
```

Sample output:

```
Test case #1:
   12   13   14
   17   18   19
--------------------
Test case #2:
   28   29
   34   35
   40   41
--------------------
Test case #3: Unable to extract requested size!
--------------------
```